

DIALOG File 351:DERWENT WPI
(c) 1999 Derwent Info Ltd. All rts. reserv.

010973796 **Image available**
WPI Acc No: 96-470745/199647
XRPX Acc No: N96-396972

**Image processor used in colour inkjet printer - in which reference
address of N dimensional table used in interpolation processing, is
calculated using third table**

Patent Assignee: CANON KK (CANO)

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Main IPC	Week
JP 8237497	A	19960913	JP 9535289	A	19950223	H04N-001/60	199647 B

Priority Applications (No Type Date): JP 9535289 A 19950223

Patent Details:

Patent	Kind	Lan	Pg	Filing Notes	Application	Patent
JP 8237497	A		15			

Abstract (Basic): JP 8237497 A

The processor performs colour conversion processing by interpolation technique using an N dimensional table. A weighting coefficient used in the interpolation processing is calculated using the first table.

A divisor is calculated using the second table. A reference address of the N dimensional table is calculated using the third table.

ADVANTAGE - Performs highly precise interpolation processing at very high speed.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-237497

(43) 公開日 平成8年(1996)9月13日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 N 1/60			H 0 4 N 1/40	D
G 0 3 G 15/01			G 0 3 G 15/01	S
G 0 6 F 17/17			G 0 6 F 15/353	
G 0 6 T 1/00			15/66	3 1 0
H 0 4 N 1/46			H 0 4 N 1/46	Z
審査請求 未請求 請求項の数14 O L (全 15 頁)				

(21) 出願番号 特願平7-35289

(22) 出願日 平成7年(1995)2月23日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 太田 享寿

東京都大田区下丸子3丁目30番2号キヤノ
ン株式会社内

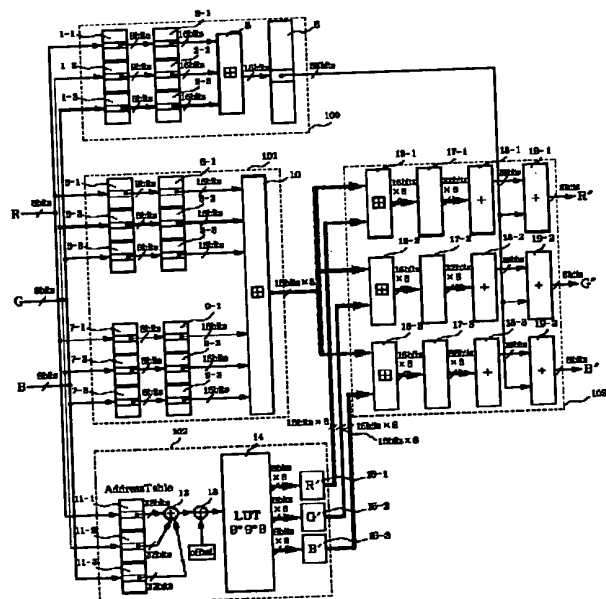
(74) 代理人 弁理士 丸島 儀一

(54) 【発明の名称】 画像処理装置及び方法

(57) 【要約】

【目的】 補間の精度が高く、格子点の取り方に柔軟性があり、なおかつ高速処理可能にすることを目的とする。

【構成】 N次元テーブルと補間処理により色変換処理を行う処理装置であって、前記補間処理に用いる重み係数を計算するための第1テーブルと、前記補間処理に用いる除数を計算するための第2テーブルと、前記補間処理に用いるN次元テーブルの参照アドレスを計算するための第3テーブルとを有することを特徴とする。



【特許請求の範囲】

【請求項1】 N次元テーブルと補間処理により色変換処理を行う処理装置であって、前記補間処理に用いる重み係数を計算するための第1テーブルと、前記補間処理に用いる除数を計算するための第2テーブルと、前記補間処理に用いるN次元テーブルの参照アドレスを計算するための第3テーブルを有することを特徴とする画像処理装置。

【請求項2】 更に、前記第1、第2、第3テーブルのうち少なくとも1つのテーブルからの出力に対して対数変換する対数変換手段を有することを特徴とする請求項1記載の画像処理装置。

【請求項3】 更に、指数変換する指数変換手段を有することを特徴とする請求項2記載の画像処理装置。

【請求項4】 前記第1のテーブルは着目点と格子点からの距離とを格納したテーブルと対数変換を行うための対数変換テーブルとで構成されることを特徴とする請求項1記載の画像処理装置。

【請求項5】 前記第1のテーブルは着目点と格子点からの距離とを対数に変換した値でもつテーブルで構成されることを特徴とする請求項1記載の画像処理装置。

【請求項6】 前記第2のテーブルは入力点を取り囲む格子点間の距離を格納したテーブルと対数変換を行うための対数変換テーブルとで構成されることを特徴とする請求項1記載の画像処理装置。

【請求項7】 前記第2のテーブルは入力点を取り囲む格子点間の距離を対数に変換した値で持つテーブルで構成されることを特徴とする請求項1記載の画像処理装置。

【請求項8】 前記第3のテーブル値を対数に変換するための対数変換テーブルを持つことを特徴とする請求項1記載の画像処理装置。

【請求項9】 前記第3のテーブルは格子点上の値を、あらかじめ対数に変換した値で持つことを特徴とする請求項1記載の画像処理装置。

【請求項10】 更に、除算を行う除算手段を有し、前記除算手段は除数が2のべき乗であるときはシフトレジスタにより除算を行うことを特徴とする請求項1記載の画像処理装置。

【請求項11】 補間処理データが以前に入力されたデータと同じであれば、前記補間処理を行わないで以前に処理されたデータを出力することを特徴とする請求項1記載の画像処理装置。

【請求項12】 0の対数を表現するのに有限のビット数で表現可能な最小の負の数を用いることを特徴とする請求項2記載の画像処理装置。

*【請求項13】 負の数を入力した場合には0を出力する指数変換テーブルを持つことを特徴とする請求項3記載の画像処理装置。

【請求項14】 N次元テーブルと補間処理により色変換処理を行う処理方法であって、前記補間処理に用いる重み係数を第1のテーブルを用いて計算し、前記補間処理に用いる除数を第2のテーブルを用いて計算し、前記補間処理に用いるN次元テーブルの参照アドレスを第3のテーブルを用いて計算することを特徴とする画像処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、テーブルを用いた色補正を行う画像処理装置及び方法に関する。

【0002】

【従来の技術】従来、この種の装置としては例えば被記録面ヘインクを付着させて記録を行うカラーインクジェットプリンタ等の色補正処理装置が知られている。これらの装置においては、例えば入力信号を入力デバイスに依存したR（レッド）、G（グリーン）、B（ブルー）の輝度信号を与え、デバイスに依存したC（シアン）、M（マゼンタ）、Y（イエロー）、K（ブラック）各色に対応する濃度信号を出力信号として得ることが必要となる。また最近では画像データのポータビリティを考慮して、このような処理系においては入力信号を一旦デバイスに非依存の色空間、例えば国際照明委員会（Commission International de l'Eclairage略してCIE）が定めるCIE 1931XYZ色空間あるいは、CIE 1976 L^{*}a^{*}b^{*}色空間に一旦変換して、さらにデバイスに依存した濃度信号へと変換するのが一般的である。このとき、色補正処理装置は図2に示すような多くの処理（処理の詳細はここでは省略する）を行わなければならない。

【0003】そこで、例えば特公昭52-16403号公報、特公昭58-16180号公報、特開昭57-208765号公報においては、あらかじめ特定の入力信号に対して、これら一連の処理を施した結果をN次元lookupアップテーブル（LUT）に格納し、このLUTを参照して補間演算を行うことにより、所望の入力信号に対応する出力信号を得る方法が提案されている。

【0004】図3は、一般的な補間方式を説明するための図であり、点P（R、G、B）が8つの格子点に含まれるような3次元の場合、格子点（R_i、G_i、B_i）が値V_j（R_i、G_i、B_i）[j=1、2…n]を持つとする点P（R、G、B）の値V_p（R、G、B）は以下の式により求められる。

【0005】

$$V_{pj}(R, G, B) = ((d_{ur} - d_r) \times (d_{ug} - d_g) \times (d_{ub} - d_b)) \times V_j(R_i, G_i, B_i)$$

3

4

$$\begin{aligned}
& \cdot B_i) + d_r \times (d_{ug} - d_g) \times (d_{ub} - d_b) \times V_j(R_{i+1}, G_i, B_i) \\
& + (d_{ur} - d_r) \times d_g \times (d_{ub} - d_b) \times V_j(R_i, G_{i+1}, B_i) + d_r \\
& \times d_g \times (d_{ub} - d_b) \times V_j(R_{i+1}, G_{i+1}, B_i) + (d_{ur} - d_r) \times (\\
& d_{ug} - d_g) \times d_b \times V_j((R_i, G_i, B_{i+1}) + d_r \times (d_{ug} - d_g) \times \\
& d_b \times V_j(R_{i+1}, G_i, B_{i+1}) + (d_{ur} - d_r) \times d_g \times d_b \times V_j(R_i \\
& , G_{i+1}, B_{i+1}) + d_r \times d_g \times d_b \times V_j(R_{i+1}, G_{i+1}, B_{i+1})) / (d_u \\
& r \times d_{ug} \times d_{ub}) \dots (\text{式1})
\end{aligned}$$

ここで(d_{ur} 、 d_{ug} 、 d_{ub})はそれぞれ各座標方向の格子点間の距離であり、(d_r 、 d_g 、 d_b)は格子点(R_i 、 G_i 、 B_i)から点P(R 、 G 、 B)までの各座標軸方向に対する距離である。

【0006】この式から、1点の値を3次元の補間によって求める場合には26回の乗算と1回の除算が必要であることがわかる。これはハードウェアで実現した場合にはその規模が大きくなり、またソフトウェアで実現した場合にはその処理時間が長くなるという欠点を持っていた。そこで、前記発明においてはハードウェアの規模縮小あるいはソフトウェアの処理スピードアップのために、利用する格子点の数を減らすことによって、演算の量を減らすという補間の方式の改良に関するアイデアが提案されている。

【0007】

【発明が解決しようとする課題】ところが、これらの方法では利用する格子点の数を減らして、演算の量を減らすことによりハードウェアの規模縮小あるいはソフトウェアの処理スピードアップを実現しようとしているために、精度の点で(式1)で示される方法よりも劣るという欠点があった。

【0008】また、図4に示されるように(式1)で示される方法では、1軸方向に関してLUTを引くためのレンジ(入力レンジ、この場合は $255 = 255 - 0 : 8 \text{ bit}$ である)が格子点の数-1(=8)で割り切れない場合には、整数を用いる限り格子点間の距離を全て等しくする事ができない。この例では244と255の間の距離が他の格子間距離よりも1小さくなっている。従って、入力レンジが1軸方向の格子点の数-1の整数倍になっていない場合には、(式1)で示される(d_{ur} 、 d_{ug} 、 d_{ub})の値は、補間に使用する格子点の位置により異なる場合があるので、 d_{ur} 、 d_{ug} 、 d_{ub} それぞれにどの値を使うのかを、使用する格子点の位置により判断しなければならないので、処理量が多くなり、処理時間が掛かるという欠点があった。

【0009】さらに、補間に利用する格子点(R_i 、 G_i 、 B_i)のアドレス、および同格子点から点P(R 、 G 、 B)までの距離を(d_r 、 d_g 、 d_b)を求める場合に、入力レンジが2のべき乗-1(例えば $255 = 2^8 - 1$)でかつ各格子点間の距離が2のべき乗(例えば $32 = 2^5$)になっていれば、(R 、 G 、 B)それぞれの値を論理演算だけで求めることが可能である。上記の例では、点P(R 、 G 、 B)を補間して求める場合に利

*用する格子点(R_i 、 G_i 、 B_i)のアドレスは(R 、 G 、 B)の上位3ビットより、また格子点(R_i 、 G_i 、 B_i)から点P(R 、 G 、 B)までの距離(d_r 、 d_g 、 d_b)は(R 、 G 、 B)の下位5ビットから簡単に求められる。ところが、各格子点間の距離が2のべき乗になっていない場合には、格子点(R_i 、 G_i 、 B_i)のアドレスおよび格子点(R_i 、 G_i 、 B_i)から点P(R 、 G 、 B)までの距離(d_r 、 d_g 、 d_b)は(R 、 G 、 B)それぞれの値を格子点間距離(d_{ur} 、 d_{ug} 、 d_{ub})によって除算し、商および余りを求めなければならない、ここでも処理量が多くなるという欠点があった。

【0010】そこで、上記2つの欠点を考慮して、格子点の数を制限して処理量を減らすという方法が試みられているが、これではN次元ルックアップテーブルの格子点の取り方に柔軟性がなくなってしまうという欠点があった。

【0011】本発明は以上のような従来の問題点に鑑みてなされたものであり、補間の精度が高く、格子点の取り方に柔軟性があり、なおかつ高速処理可能にすることを目的とする。

【0012】

【課題を解決するための手段及び作用】本発明は上述の目的を達成するために、N次元テーブルと補間処理により色変換処理を行う処理装置であって、前記補間処理に用いる重み係数を計算するための第1テーブルと、前記補間処理に用いる除数を計算するための第2テーブルと、前記補間処理に用いるN次元テーブルの参照アドレスを計算するための第3テーブルを有することを特徴とする。

【0013】

【実施例】

(実施例1)以下、図面を参照して本発明の実施例を詳細に説明する。

【0014】図1は、本発明の第1の実施例に関わる画像処理装置の構成を説明するブロック図である。この図においてR、G、Bで示されるものは入力データで、Red、Green、Blue各々8ビットデータが入力される。入力されたR、G、Bデータはそれぞれ、除数計算ブロック100、重み係数計算ブロック101、LUT読み出しブロック102に入力される。さらに除数計算ブロック100の出力、および重み係数計算ブロック101の出力、さらにLUT読み出しブロック102

* 50

5

の出力は、補間処理ブロック103に入力されて補間処理が施され、それぞれ8bitのデータR'、G'、B'が出力される。

【0015】まず、除数計算ブロック100は、入力されたR、G、Bの値をLUT14上で座標値とする点Pを取り囲む8つの格子点を作る直方体の体積を計算するためのブロックである。このブロックは、入力されたR、G、Bの値から8つの格子点を作る直方体の各軸方向の長さを求めるためのテーブル1-1、1-2、1-3、同テーブルで求めた値の対数を求めるための対数テーブル2-1、2-2、2-3、および同対数テーブルで求めた値を足し合わせるための加算手段3、および加算手段3によって求められた値をリニアな値に戻すための指数テーブル5からなる。

【0016】テーブル1-1、1-2、1-3は入力されたR、G、Bの値から点Pを取り囲む直方体の各軸方向の長さを求めるためのもので、256個のインデックスを持つ8bitのテーブルで構成されている。本実施例においては、入力データはR、G、Bの各々8bitを想定しているので、本テーブルを参照することにより各々の入力に対応した8bitのデータが出力される。図5は本テーブルの内容を図示したものであり、本実施例においては格子点の数は図4に示したように各軸方向に9つあることを想定しているので、格子点距離が等しくないものをも反映したものとなっている。

【0017】テーブル2-1、2-2、2-3はそれぞれテーブル1-1、1-2、1-3によって出力された値を、2を底とする対数に変換するためのテーブルで、256個のインデックスを持つ16bitのテーブルで構成されている。本実施例においては、テーブル1-1、1-2、1-3から出力されるデータは各々8bitを想定しているので、本テーブルを参照することにより各々の入力に対応した16bitのデータが出力される。図6はこのテーブルの内容を図示したもので、その値としては次式で与えられる値があらかじめ代入されている。

【0018】
対数値 = $\log_2(\text{入力値}) \times 1024 \dots (\text{式2})$

これは、入力値の2を底とする対数を求め、その値を上位1bitを符号、下位10bitを少数部、残り5bitを整数部とする16bitの固定少数点値として表すための式である。なお本テーブルにより $\log_2(0)$ を表すために、本実施例においては最も大きな負の数(-32768)を採用している。

【0019】加算手段3はテーブル2-1、2-2、2-3によって出力された対数値を加算するためのものであり、図7に示すように加算器3-1、3-2およびシフトレジスタ3-3から構成される。加算器3-1、3-2はテーブル2-1、2-2、2-3の対数出力を加算するための16bit加算器である。シフトレジスタ

6

3-3は加算器3-2から出力された16bitデータを右に2bitシフトし、上位1bitを符号、下位8bitを少数部、残り5bitを整数部とする14bitデータを得るためのシフトレジスタである。よって、加算演算は16bitデータに基づき高精度に行う一方、シフトレジスタにより14bitデータに変換することによりテーブル容量を減らすことができる。

【0020】テーブル5は加算手段3によつと求められた値をリニアな値に戻すための指数テーブルであり、8192(=2¹³)個のインデックスを持つ32bitのテーブルにより構成されている。本実施例においては、加算手段3から出力されるデータを入力値とし、その値が負の場合には本テーブルは参照されないで直ちに0が出力される。一方、同値が負でない場合には本テーブルを参照することにより入力に対応した32bitのデータが出力される。図8はこのテーブルを図示したもので、その値として次式で与えられる値があらかじめ代入されている。

【0021】

【外1】

$$\text{べき乗値} = 2^{(\text{入力値} / 256) \dots (\text{式3})}$$

なお、入力値を256で割っているのは、入力値14bitの下位8bitが少数部分となっているからである。

【0022】以上により、除数計算ブロック100は、R、G、Bの値を入力することにより、同値をLUT14上で座標値とする点Pを取り囲む8つの格子点を作る直方体の体積を出力する。通常体積を計算する場合には2回の乗算が必要であるが、本実施例によればルックアップテーブルを利用することにより2回の足し算に帰着された。

【0023】次に、重み係数計算ブロック101は、入力されたR、G、Bの値をLUT14上で座標値とする点Pの持つ値R'、G'、B'の値を補間処理により求めるために、同点Pを取り囲む8つの格子点に与えるべき重み係数を計算するためのブロックである。このブロックは、入力されたR、G、Bの値から図3で示されるdr、dg、dbおよびdur-dr、dug-dg、dub-dbを求めるためのテーブル6-1、6-2、6-3および7-1、7-2、7-3と同テーブルで求めた値の対数を求めるための対数テーブル8-1、8-2、8-3および9-1、9-2、9-3と、同対数テーブルで求めた値を足し合わせて8つの格子点に対する重み係数を求めるための加算手段10からなる。

【0024】テーブル6-1、6-2、6-3は、入力されたR、G、Bの値に対応して図3で示されるdr、dg、dbを求めるためのもので、256個のインデックスを持つ8bitのテーブルで構成されている。本実施例においては、入力データはR、G、B各々8bitを想定しているので、本テーブルを参照することにより

7

各々の入力に対応した8bitのデータが出力される。図9は本テーブルの内容を図示したものであり、本実施例においては格子点の数は各軸方向に9つあることを想定している。図4で示される格子点から入力値までの距離が格納されている。同様にテーブル7-1、7-2、7-3は、入力されたR、G、Bの値に対応した d_{ur-dr} 、 d_{ug-dg} 、 d_{ub-db} を求めるためのもので、256個のインデックスを持つ8bitのテーブルで構成されている。本実施例においては、入力データはR、G、B各々8bitを想定している。本テーブルを参照することにより各々の入力に対応した8bitのデータが出力される。図10は本テーブルの内容を図示したものであり、本テーブル7-1、7-2、7-3と前記テーブル6-1、6-2、6-3とを入力値毎に合計したものはちょうどテーブル1-1、1-2、1-3に等しくなっている。

【0025】テーブル8-1、8-2、8-3および9-1、9-2、9-3はそれぞれテーブル6-1、6-2、6-3および7-1、7-2、7-3によって出力された値の、2を底とする対数を得るためのテーブルで、256個のインデックスを持つ16bitのテーブルで構成されている。これは、2-1、2-2、2-3で示された対数テーブルと同じものである。

【0026】加算手段10はテーブル8-1、8-2、8-3および9-1、9-2、9-3によって出力された対数値を、組み合わせて加算するためのものであり、加算器10-1、10-2、10-3、10-4、10-5、10-6、10-7、10-8、10-9、10-10、10-11、10-12によって構成される。これらの加算器10-1~12は16bit加算器であり、テーブル8-1、8-2、8-3および9-1、9-2、9-3の出力を組み合わせて加算することにより8つの格子点に対応した重み係数を算出する。これを図示したものが図11である。このような構成にすることで、通常12回の乗算がルックアップテーブルを利用することにより12回の足し算に帰着された。

【0027】さらに、LUT読み出しブロック102は、入力された入力されたR、G、Bの値によって図12に示したLUT14を参照し、8つの格子点にある、あらかじめ補正処理を施されたR'、G'、B'の値を得るためのブロックである。このブロックは、入力されたR、G、Bの値から図3で示される格子点(Ri、Gi、Bi)のアドレスを求めるためのテーブルを11-1、11-2、11-3と、加算器12、13とLUT14および対数変換テーブル15-1、15-2、15-3からなる。

【0028】図12は本実施例におけるLUT14の構成を示したものである。R、G、Bそれぞれ9個、合計792(=9×9×9)個の格子点に対して、あらかじめ色補正処理を施した値R'、G'、B'各々8bit

8

が図のように格納されている。

【0029】テーブル11-1、11-2、11-3は入力されたR、G、Bの値からLUT14を参照するためのアドレスを計算するためのテーブルであり、それぞれ256個のインデックスを持つ32bitのテーブルで構成されている。本実施例においては、入力データはR、G、B各々8bitを想定している。本テーブルを参照することにより各々の入力に対応した32bitのデータが出力される。図13はこのテーブルの内容を図示したものである。

【0030】加算手段12はテーブル11-1、11-2、11-3によって出力された値を加算するためのものであり、これにより直ちにLUT14を参照すべき開始アドレスが求まる。

【0031】加算手段13は加算手段12によって出力されたLUT14の参照開始アドレスに対してオフセット値を加えることにより、8つの格子点のR'、G'、B'各色に対する各々のアドレスを得るものである。本実施例においては、LUT14の格子点のR、G、B各々9個なので、8つの格子点(Ri、Gi、Bi)、(Ri+1、Gi、Bi)、(Ri、Gi+1、Bi)、(Ri+1、Gi+1、Bi)、(Ri、Gi、Bi+1)、(Ri+1、Gi、Bi+1)、((Ri、Gi+1、Bi+1)、((Ri+1、Gi+1、Bi+1)に対する、加算手段12の出力からのオフセット値はそれぞれ0、3、27(=9×3)、30(=27+3)、243(=9×9×3)、246(=243+3)、270(=243+27)、273(=270+3)である。またR'、G'、B'各色に対するオフセットはそれぞれ上記オフセット値に0、1、2を加えたものとなる。

【0032】加算手段13から出力されたLUT参照アドレスにより、LUT14を参照し、R'、G'、B'それぞれ8個、合計24個の8bitデータが出力される。これらのデータは、各色毎に、対数変換テーブル15-1、15-2、15-3に入力され対数に変換されて出力される。

【0033】対数変換テーブル15-1、15-2、15-3はLUT14によって各色毎に出力された値の、2を底とする対数を得るためのテーブルで、256個のインデックスを持つ16bitのテーブルで構成されている。これは、2-1、2-2、2-3で示された対数テーブルと同じものであり、1テーブルにつき8個のデータが出力される。

【0034】最後に補間処理ブロック103は、除数計算ブロック100の出力、および重み係数計算ブロック101の出力、およびLUT読み出しブロック102の出力を入力として、補間処理を行い、補間された値R''、G''、B''出力するブロックである。このブロックは、重み係数計算ブロック101の出力およびLUT読み出しブロック102の出力を組み合わせて加算する

10

20

30

40

50

加算手段16-1、16-2、16-3、同加算器からの出力をリニアな値に戻すための指数テーブル17-1、17-2、17-3、同指数テーブルから出力を各色毎に足し合わせるための加算手段18-1、18-2、18-3、および同加算器の出力を除数計算ブロック100の出力で割り算するための除算手段19-1、19-2、19-3から構成される。

【0035】加算手段16-1、16-2、16-3は重み係数計算ブロック101の出力およびLUT読み出しブロック102の出力を組み合わせて加算する加算手段であり、その詳細は図14で示される。

【0036】図14は加算手段16-1、16-2、16-3の詳細を示した図であり、重み係数計算ブロック101からの8つの出力、

$$\begin{aligned} & (d_{ur}-dr) \times (d_{ug}-dg) \times (d_{ub}-db) \\ & dr \times (d_{ug}-dg) \times (d_{ub}-db) \\ & (d_{ur}-dr) \times dg \times (d_{ub}-db) \\ & dr \times dg \times (d_{ub}-db) \\ & (d_{ur}-dr) \times (d_{ug}-dg) \times db \\ & dr \times (d_{ug}-dg) \times dg \\ & (d_{ur}-dr) \times dg \times db \\ & dr \times dg \times db \end{aligned}$$

に対して、LUT読み出しブロック102からの1色分の出力

$$\begin{aligned} & V(R_i, G_i, B_i) \\ & V(R_{i+1}, G_i, B_i) \\ & V(R_i, G_{i+1}, B_i) \\ & V(R_{i+1}, G_{i+1}, B_i) \\ & V(R_i, G_i, B_{i+1}) \\ & V(R_{i+1}, G_i, B_{i+1}) \\ & V(R_i, G_{i+1}, B_{i+1}) \\ & V(R_{i+1}, G_{i+1}, B_{i+1}) \end{aligned}$$

〔但しVはR、G、Bのいずれか〕を16bit加算器によってそれぞれ加算する加算器161-1、161-2、161-3、161-4、161-5、161-6、161-7、161-8、と、同加算器からの出力を右に2bitシフトし、上位1bitを符号、下位8bitを少数部、残り5bitを整数部とする14bitデータを得るためのシフトレジスタ162-1、162-2、162-3、162-4、162-5、162-6、162-7、162-8から構成されている。

【0037】このような構成にすることで、通常8回の乗算がここでも8回の足し算に帰着された。

【0038】テーブル17-1、17-2、17-3は加算手段16-1、16-2、16-3によって求められた各色8個の値をリニアな値に戻すための指数テーブルであり、 $8192 (= 2^{13})$ 個のインデックスを持つ32bitのテーブルにより構成されており、これはテーブル5と同様のものである。本実施例においては、加算手段16-1、16-2、16-3から出力されるデ

ータ入力値とし、その値が負の場合には本テーブルは参照されないで直ちに0が出力され、一方、同値が負でない場合には本テーブルを参照することにより入力に対応した32bitのデータが出力される。

【0039】加算手段18-1、18-2、18-3、はテーブル17-1、17-2、17-3から出力される各色8個のデータを加算するための加算手段であり、その詳細は図15で示される。

【0040】図15は加算手段18-1、18-2、18-3の詳細を示した図であり、本加算手段はテーブル17-1、17-2、17-3から出力される各色8個のデータ、

$$(d_{ur}-dr) \times (d_{ug}-dg) \times (d_{ub}-db) \times V(R_i, G_i, B_i)$$

$$dr \times (d_{ug}-dg) \times (d_{ub}-db) \times V(R_{i+1}, G_i, B_i)$$

$$(d_{ur}-dr) \times dg \times (d_{ub}-db) \times V(R_i, G_{i+1}, B_i)$$

$$dr \times dg \times (d_{ub}-db) \times V(R_{i+1}, G_{i+1}, B_i)$$

$$(d_{ur}-dr) \times (d_{ug}-dg) \times db \times V(R_i, G_i, B_{i+1})$$

$$dr \times (d_{ug}-dg) \times db \times V(R_{i+1}, G_i, B_{i+1})$$

$$(d_{ur}-dr) \times dg \times db \times V(R_i, G_{i+1}, B_{i+1})$$

$$dr \times dg \times db \times V(R_{i+1}, G_{i+1}, B_{i+1})$$

〔但しVはR、G、Bのいずれか〕を32bit加算器181-1、181-2、181-3、181-4、181-5、181-6、181-7によって加算して出力する。

【0041】除算手段19-1、19-2、19-3、は加算手段18-1、18-2、18-3から出力される各色32bitのデータを除数計算ブロック100の32bit出力によって除算するため除算手段である。これにより、補間処理されたR'、G'、B' 各々8bitのデータを出力される。

【0042】以上により、R、G、B各々8bitのデータが入力に対して(式1)で表される補間処理が複数のLUTを用いることにより乗算無しでかつ、高速に実行され、R'、G'、B' それぞれ8bitのデータが出力される。

【0043】(実施例2) 図16は本発明の第2の実施例に係わる画像処理装置の構成を説明するブロック図である。

【0044】本実施例においては、第1の実施例ではR、G、B3入力に対して、R'、G'、B' の3出力を返していたが、LUT14にC、M、Y、K4色分のデータを持ち、補間処理ブロック103を構成する加算手段、指数テーブル、加算手段、除算手段を4色分持つ

11

ことにより、図2で説明した色補正処理の全行程を1つのLUTを参照することにより可能としている。

【0045】(実施例3) 図17は本発明の第3の実施例に係わる画像処理装置の構成を説明するブロック図である。

【0046】第1の実施例ではR、G、B3入力に対して、R'、G'、B'の3出力を返していたが、本実施例においては、C、M、Y、K4入力に対してR'、G'、B'の3出力を返すようになっている。これは、*

$$V_{pj}(C, M, Y, K) = \\ ((d_{uc}-dc) \times (d_{um}-dm) \times (d_{uy}-dy) \times (d_{uk}-dk) \\ \times V_j(C_i, M_i, Y_i, K_i) + dc \times (d_{um}-dm) \times (d_{uy}-dy) \times (d_{uk}-dk) \times V_j(C_{i+1}, M_i, Y_i, K_i) + (d_{uc}-dc) \times dm \times (d_{uy}-dy) \times (d_{uk}-dk) \times V_j(C_i, M_{i+1}, Y_i, K_i) + dc \times dm \times (d_{uy}-dy) \times (d_{uk}-dk) \times V_j(C_{i+1}, M_{i+1}, Y_i, K_i) + (d_{uc}-dc) \times (d_{um}-dm) \times (d_{uk}-dk) \times dy \times V_j(C_i, M_i, Y_{i+1}, K_i) + dc \times (d_{um}-dm) \times (d_{uk}-dk) \times dy \times V_j(C_{i+1}, M_i, Y_{i+1}, K_i) + (d_{uc}-dc) \times dm \times dy \times (d_{uk}-dk) \times V_j(C_i, M_{i+1}, Y_{i+1}, K_i) + dc \times dm \times dy \times (d_{uk}-dk) \times V_j(C_{i+1}, M_{i+1}, Y_{i+1}, K_i) + (d_{uc}-dc) \times (d_{um}-dm) \times (d_{uy}-dy) \times dk \times V_j(C_i, M_i, Y_i, K_{i+1}) + dc \times (d_{um}-dm) \times (d_{uy}-dy) \times dk \times V_j(C_{i+1}, M_i, Y_i, K_{i+1}) + (d_{uc}-dc) \times dm \times (d_{uy}-dy) \times dk \times V_j(C_i, M_{i+1}, Y_i, K_{i+1}) + dc \times dm \times (d_{uy}-dy) \times dk \times V_j(C_{i+1}, M_{i+1}, Y_i, K_{i+1}) + (d_{uc}-dc) \times (d_{um}-dm) \times dy \times dk \times V_j(C_i, M_i, Y_{i+1}, K_{i+1}) + dc \times (d_{um}-dm) \times dy \times dk \times V_j(C_{i+1}, M_i, Y_{i+1}, K_{i+1}) + (d_{uc}-dc) \times dm \times dy \times dk \times V_j(C_i, M_{i+1}, Y_{i+1}, K_{i+1}) + dc \times dm \times dy \times dk \times V_j(C_{i+1}, M_{i+1}, Y_{i+1}, K_{i+1})) / (d_{uc} \times d_{um} \times d_{uy} \times d_{uk}) \dots (式3)$$

ここで(d_{uc}、d_{um}、d_{uy}、d_{uk})はそれぞれ各座

標軸方向の格子点間の距離であり、(dc、dm、dy、dk)は格子点(C_i、M_i、Y_i、K_i)から点P(C、M、Y、K)までの各座標軸方向に対する距離である。

【0049】(式3)で表される補間方式を実現するために、実施例3においては、除数計算ブロック100、重み係数計算ブロック101、LUT読み出しブロック102のそれぞれにC、M、Y、Kの4色が入力されており、それらの構成は以下のように変更されている。

【0050】除数計算ブロック100においてはテーブル1-4、対数テーブル2-4が追加されている。また重み係数計算ブロック101においては、テーブル6-4および7-4、対数テーブル8-4および9-4が追加されている。さらに、LUT読み出しブロック102においてはテーブル11-4が追加されている。

【0051】以上によりC、M、Y、Kの4入力に対してR'、G'、B'の3出力を返すことを可能としたものである。

【0052】(実施例4) 図18は本発明の第4の実施例に係わる画像処理装置を構成する除算手段を説明する※50

12

*例えば、出力デバイス依存のC、M、Y、Kデータをモニタ上でプレビューを行う場合に有効なものである。

【0047】点P(C、M、Y、K)が16個の格子点に含まれるような4次元の場合、格子点(C_i、M_i、Y_i、K_i)が値V_j(C_i、M_i、Y_i、K_i) [j=1、2、…n]を持つとすると点P(C、M、Y、K)の値V_{pj}(C、M、Y、K)は以下の式により求められる。

【0048】

30※ブロック図である。

【0053】本実施例においては、実施例1にあった対数変換テーブル2-1、2-2、2-3、8-1、8-2、8-3、9-1、9-2、9-3、15-1、15-2、15-3を省略した代わりに、テーブル1-1、1-2、1-3、6-1、6-2、6-3、7-1、7-2、7-3およびLUT14にはあらかじめ対数に変換された16bitの値が格納されている。こうすれば、ルックアップテーブルを2段に参照する必要がなくなりより高速な処理が可能となる。

【0054】(実施例5) 図19は本発明の第5の実施例に係わる画像処理装置を構成を説明するブロック図である。

【0055】補間処理ブロックの除算手段は、第1の実施例においては1つの除算器であったが、本実施例においては除算器19'-1、シフトレジスタ19'-2、比較器19'-3およびセレクト19'-4より構成されている。これは除算手段に入力された除数が2のべき乗に等しくなっている場合には、除算演算はシフトレジスタにより高速に実行が可能であるので、除数が2のべき乗であるかどうかを比較器19'-3により判断し

13

て、そうであればシフトレジスタを、そうでなければ除算器を利用して計算を行うことにより高速な処理を可能としたものである。

【0056】(実施例6)図20は本発明の第6の実施例に係わる画像処理装置の構成を説明するブロック図である。

【0057】本実施例においては、入力データは記憶手段(1)20、および比較手段21にも入力されている。これによりR、G、B入力データは記憶手段20に記憶され、次に入力されるデータと比較手段21によって比較される。もしも、記憶手段21からの出力、すなわち1画素前に入力されたデータと現在入力されたデータが等しければ、記憶手段(2)22に記憶されている1画素前の出力データがセクタ23によって出力され、実施例1で示したような処理は実行されない。また等しくないときには、通常の処理が施されたデータがセクタ23によって選択され出力される。これにより、同じデータが続けて入力された場合には、一連の処理を省くことにより高速な処理を可能としたものである。

【0058】以上により、従来の補間方式よりも精度が高く、LUTの構成に柔軟性の高くなかつ高速な補間方式が実現可能となった。

【0059】なお、ここで述べた実施例においては入力データの1チャンネルあたりのビット数は8bitであったが、これに限定されるものではない。

【0060】また実施例1、2および3からわかるように、本発明においては入力のチャンネル数、および出力のチャンネル数は3あるいは4に限定されない。異なった入力のチャンネル数に柔軟に対応するためには、入力のチャンネル数に応じて、除数計算ブロック100におけるテーブル1および2、重み係数計算ブロック101におけるテーブル6、7、8、9、LUT読み出しブロック102におけるテーブル11を、入力チャンネル数分持てば良い。同様に、異なった出力のチャンネル数に柔軟に対応するためには、出力のチャンネル数に応じて、LUT読み出しブロック102におけるLUT14の格納データ、補間処理ブロック103における加算手段16、テーブル17、加算手段18および除算手段19を、出力のチャンネル数分持てば良い。従って、C、M、Y、K4入力、C'、M'、Y'、K'4出力や、R、G、B3入力GRAY1出力といった様々な組み合わせが実現可能である。

【0061】また、実施例においてはLUT14の各軸方向に対する格子点の数を9個で説明したが、この数は9に限らない。除数計算ブロック100におけるテーブル1、重み係数計算ブロック101におけるテーブル6、7LUT読み出しブロック102におけるテーブル11の内容を、格子点の数に見合ったものにしておけば格子点の数は自由に選択することができる。また、これらの実施例の構成を用いれば、格子点の数は各軸方向で

14

異なってもかまわない。逆に、各軸方向で同じ数の格子点を用いるならば、除数計算ブロック100におけるテーブル1、重み係数計算ブロック101におけるテーブル6、7、は1つにまとめることも可能である。

【0062】以上説明したように、本実施例によれば、N次元ルックアップテーブルと補間処理により色変換処理をおこなう処理装置において、補間に用いる重み係数を計算するためのルックアップテーブルと、補間に用いる除数を計算するためルックアップテーブルと補間に用いるN次元ルックアップテーブルの参照アドレス計算するためのルックアップテーブルと、指数変換を行うための指数テーブルとを備え、補間処理の計算には対数値を使って高速に演算を行うことにより、精度が高く、高速で、しかもN次元ルックアップテーブルの構成に柔軟性の高い補間方式が実現可能となった。

【0063】

【発明の効果】本発明によれば、シンプルで精度が良く、なおかつN次元ルックアップテーブルの構成に柔軟性の高い、高速に補間処理を行うことができる。

【0064】また、他の発明によれば対数変換を含めたテーブルを用いるので高速に補間処理を行うことができる。

【図面の簡単な説明】

【図1】第1の実施例の画像処理装置の構成の例を示すブロック図。

【図2】色変換処理の流れの例を示すフローチャート。

【図3】補間方式の例を説明するための模式図。

【図4】格子点と格子点間の距離の例を示す模式図。

【図5】格子点間の距離を求めるためのテーブルの例を示す模式図。

【図6】対数変換テーブルの例を示す模式図。

【図7】加算手段3の構成の例を示すブロック図。

【図8】指数テーブルの例を示す模式図。

【図9】テーブル6の例を示す模式図。

【図10】テーブル7の例を示す模式図。

【図11】加算手段10の構成の例を示すブロック図。

【図12】3次元ルックアップテーブルの例を示す模式図。

【図13】テーブル11-1、11-2、11-3の例を示す模式図。

【図14】加算手段16の構成の例を示すブロック図。

【図15】加算手段18の構成の例を示すブロック図。

【図16】第2の実施例の画像処理装置の構成の例を示すブロック図。

【図17】第3の実施例の画像処理装置の構成の例を示すブロック図。

【図18】第4の実施例の画像処理装置の構成の例を示すブロック図。

【図19】第5の実施例における除算手段の構成の例を示すブロック図。

15

16

【図20】第6の実施例の画像処理装置の構成の例を示すブロック図。

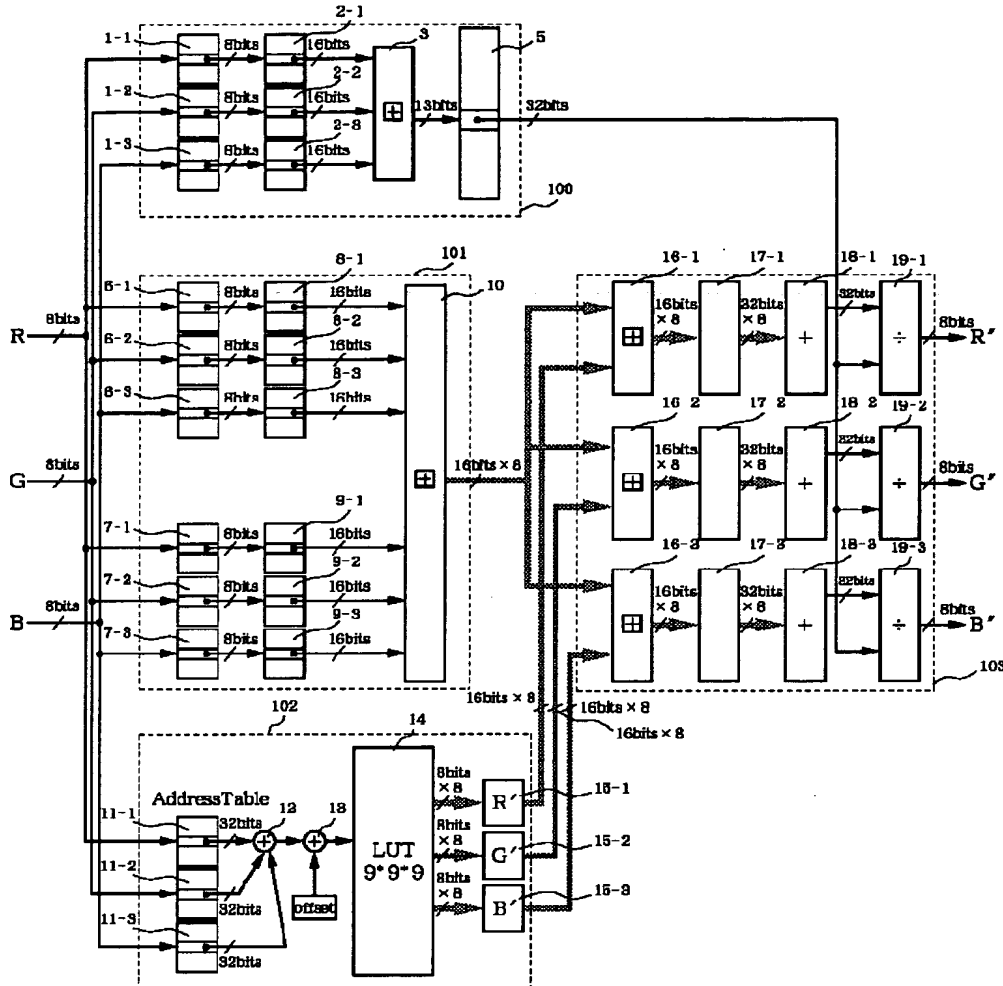
【符号の説明】

100 除数計算ブロック

101 重み係数計算ブロック

102 LUT読み出しブロック

【図1】



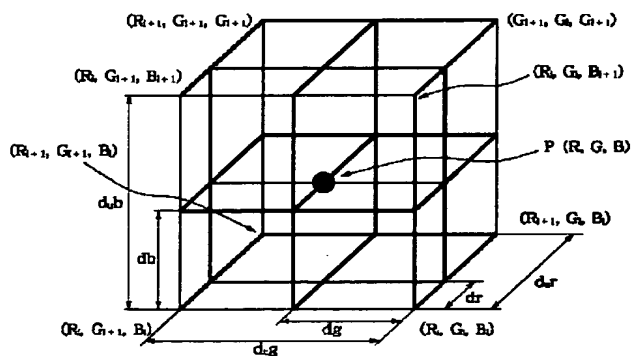
【図5】

入力値	格子点の距離
0	32
1	32
2	32
3	32
...	...
30	32
31	32
32	32
33	32
34	32
...	...
221	32
222	32
223	32
224	31
225	31
226	31
...	...
253	31
254	31
255	31

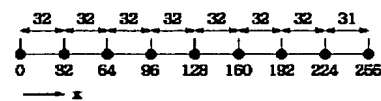
【図6】

入力値	対数値
0	32768
1	0
2	1024
3	1628
4	2048
5	2378
6	2647
...	...
253	8175
254	8180
255	8188

【図3】

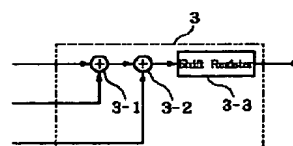


【図4】

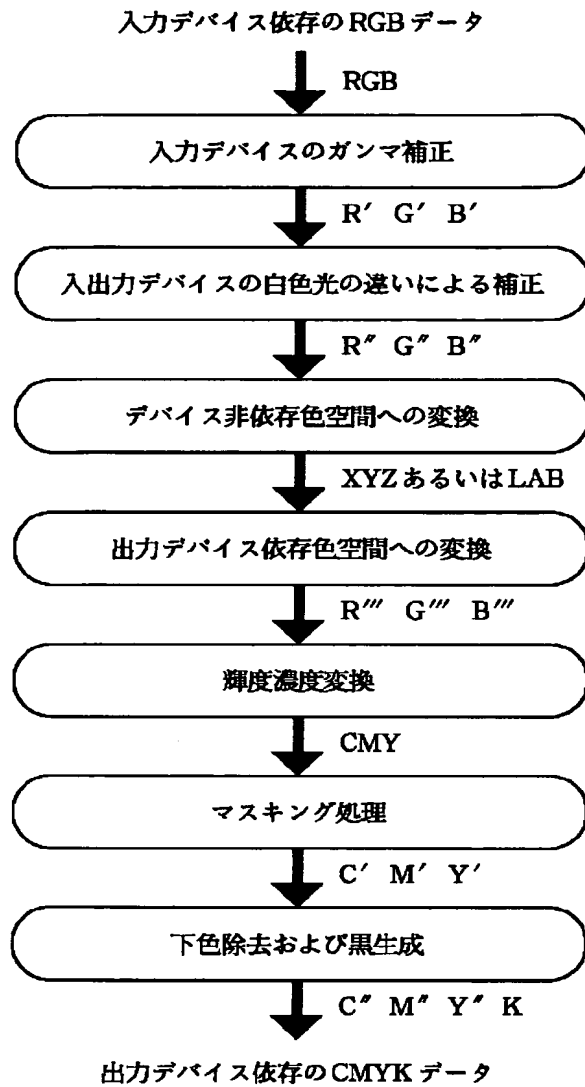


● 格子点

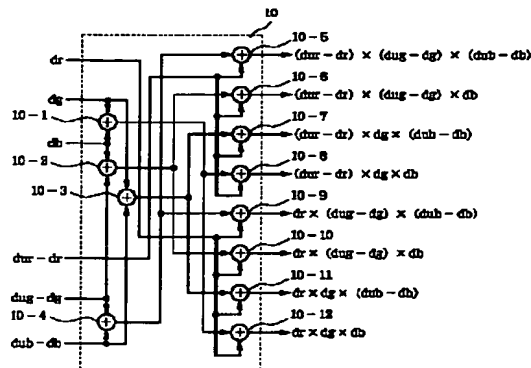
【図7】



【図2】



【図11】



【図8】

入力値	べき乗値
0	1
1	1
2	1
3	1
4	1
5	1
6	1
...	...
149	1
150	2
151	3
...	...
8189	4280221365
8190	4271771996
8191	4289363945

【図9】

入力値	内容
0	0
1	1
2	2
3	3
...	...
30	30
31	31
32	0
33	1
34	2
...	...
221	29
222	30
223	31
224	0
225	1
226	2
...	...
253	29
254	30
255	31

【図10】

入力値	内容
0	32
1	31
2	30
3	29
...	...
30	2
31	1
32	33
33	31
34	30
...	...
221	3
222	2
223	1
224	31
225	30
226	29
...	...
253	2
254	1
255	0

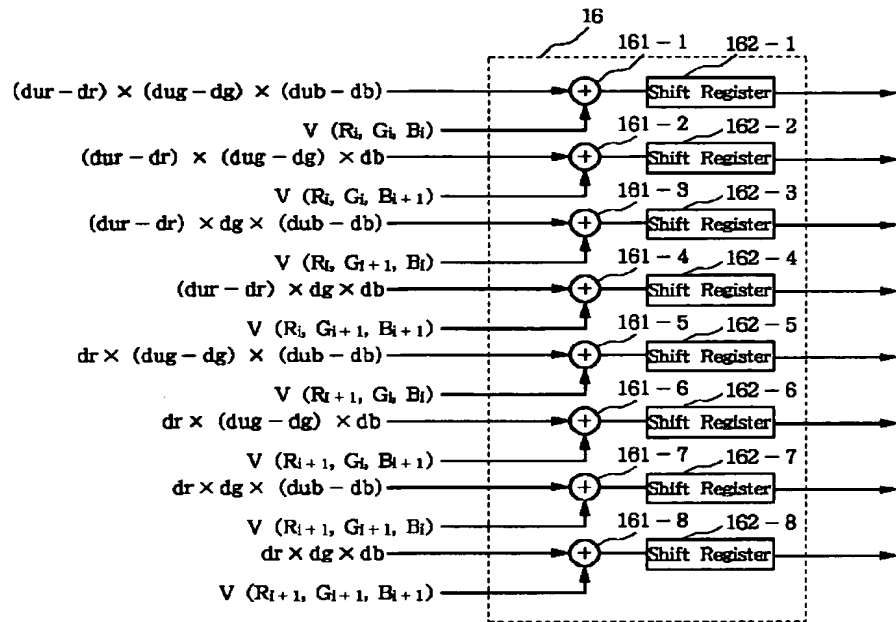
【図12】

アドレス	R	G	B	R'	G'	B'
0	0	0	0	x	x	x
3	0	0	1	x	x	x
6	0	0	2	x	x	x
9	0	0	3	x	x	x
...
21	0	0	7	x	x	x
24	0	0	8	x	x	x
27	0	1	0	x	x	x
30	0	1	1	x	x	x
...
48	0	1	7	x	x	x
61	0	1	8	x	x	x
64	0	2	0	x	x	x
67	0	2	1	x	x	x
...
237	0	8	7	x	x	x
240	0	8	8	x	x	x
243	1	0	0	x	x	x
246	1	0	1	x	x	x
...
2178	8	8	8	x	x	x
2181	8	8	7	x	x	x
2184	8	8	8	x	x	x

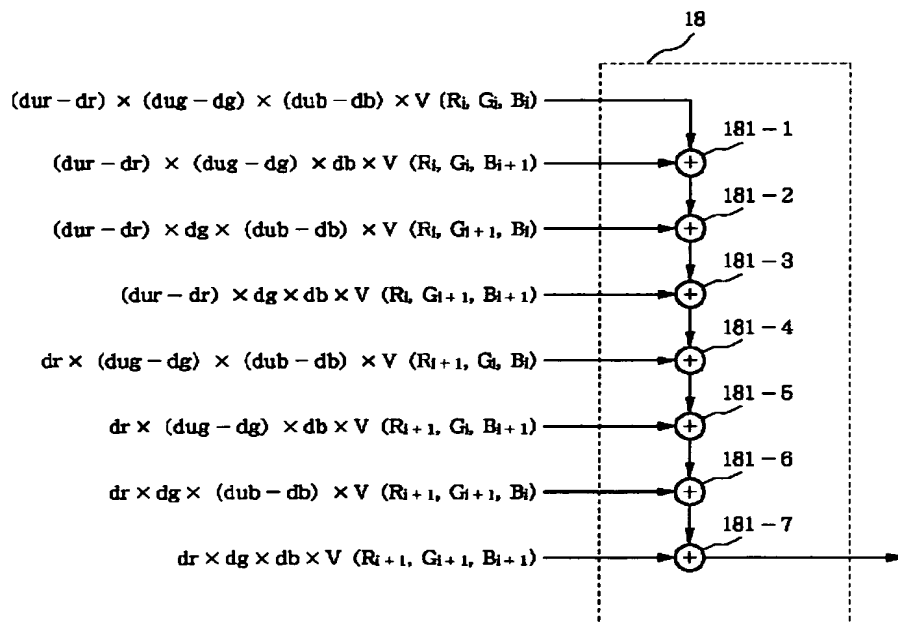
【図13】

入力値	11-1	11-2	11-3
0	0	0	0
1	0	0	3
2	0	0	6
3	0	0	9
...
7	0	0	21
8	0	0	24
9	0	27	0
10	0	27	3
...
16	0	27	21
17	0	27	24
18	0	54	0
19	0	54	3
...
79	0	216	21
80	0	216	24
81	243	0	0
82	243	0	3
...
253	1944	216	18
254	1944	216	21
255	1944	216	24

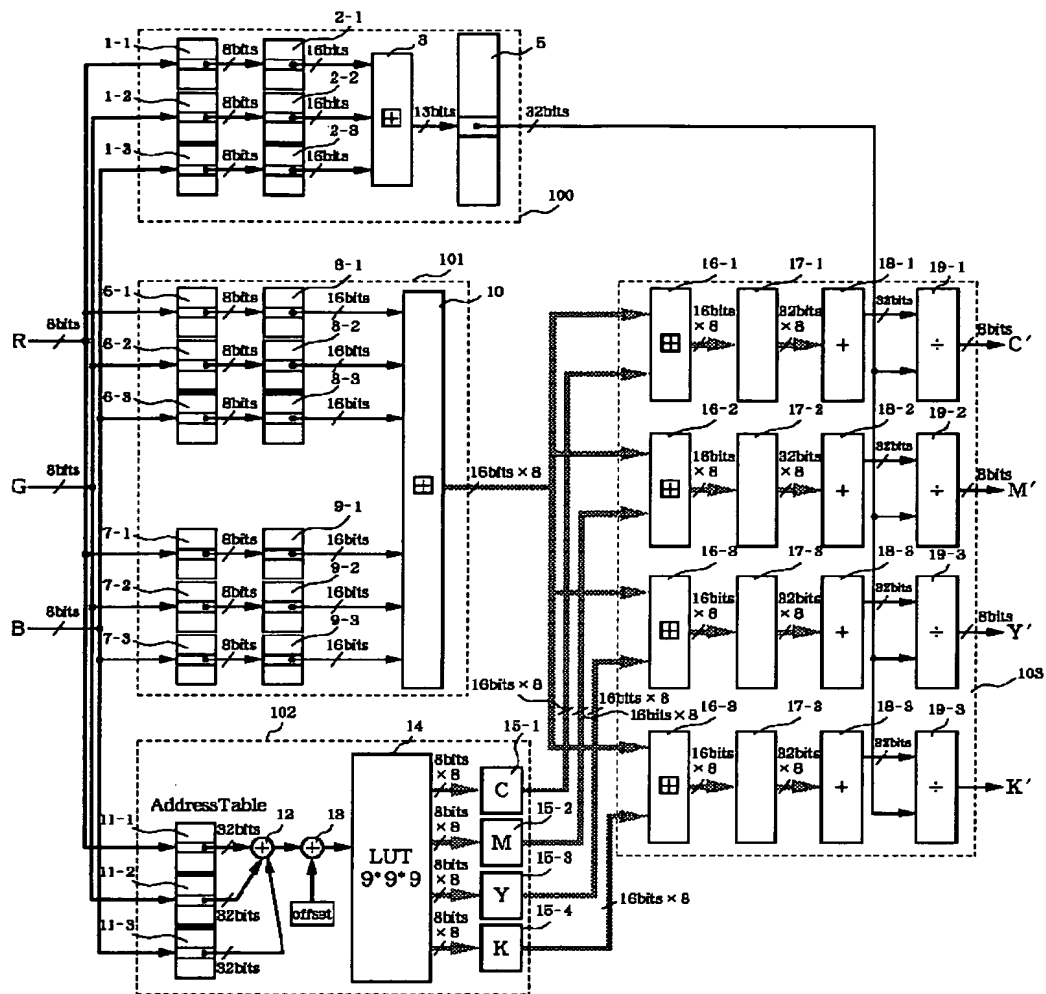
【図14】



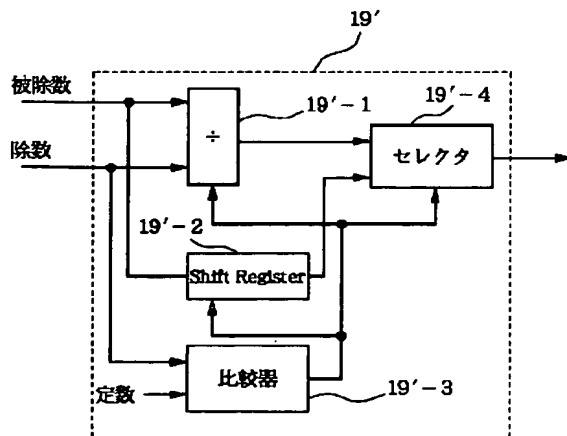
【図15】



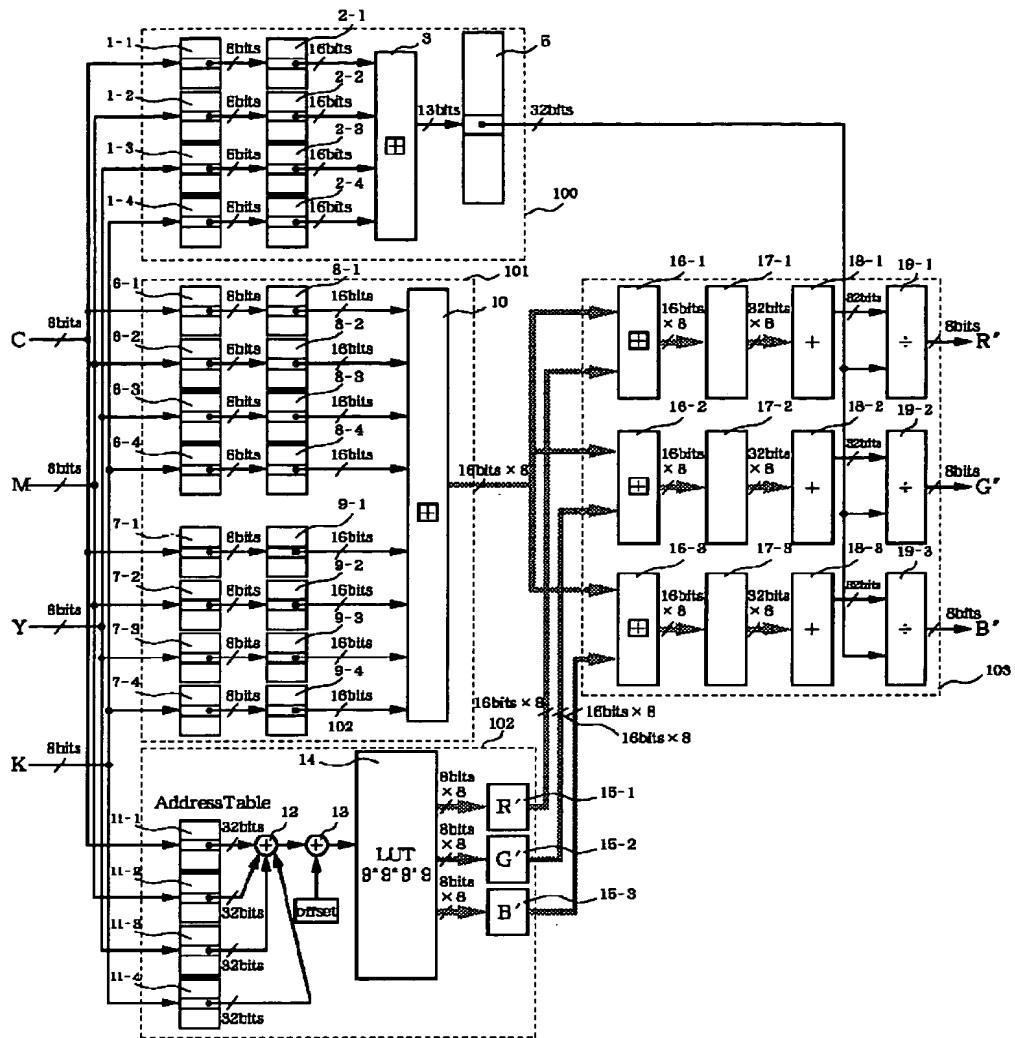
【図16】



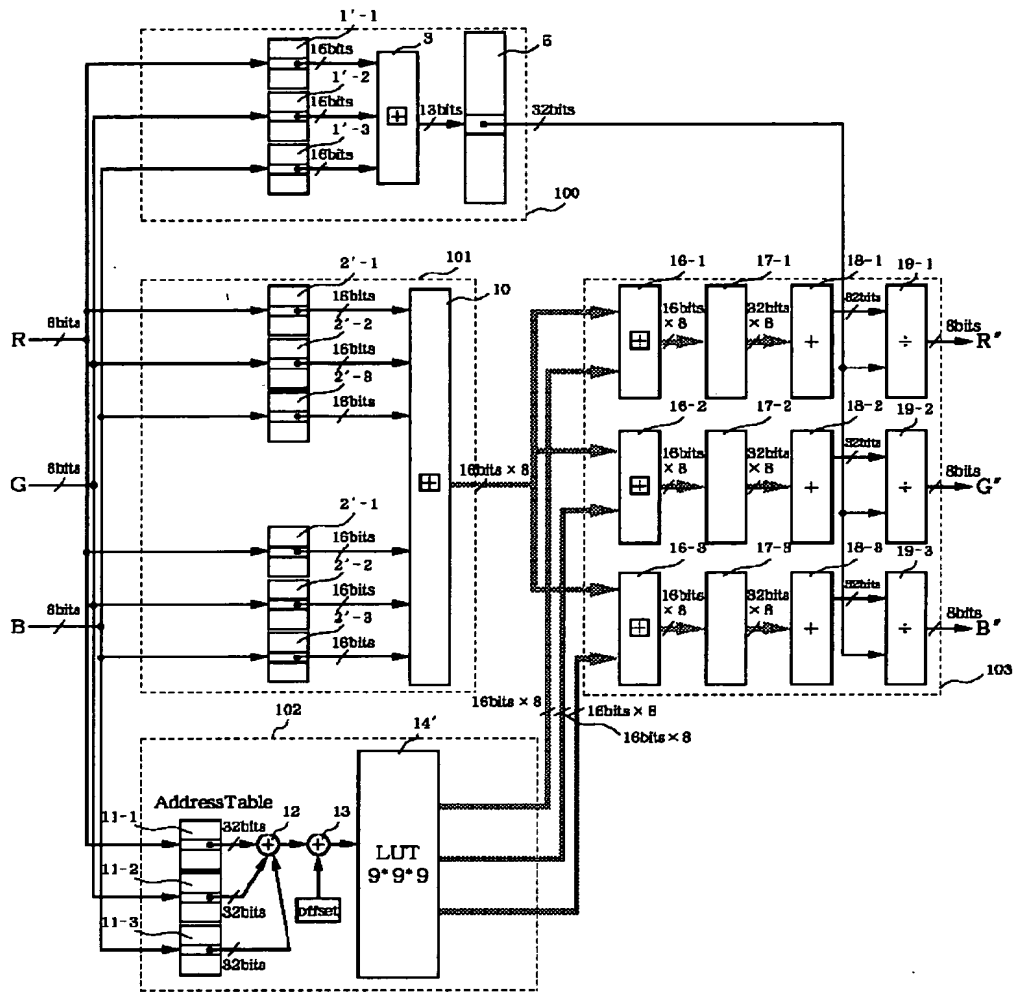
【図19】



【図17】



【図18】



【図20】

